Energy, Mines and Resources Canada | Énergie, Mines et Ressources Canada

CANMET

Canada Centre for Mineral and Energy Technology | Centre canadien de la technologie des minéraux et de l'énergie

# PROGRAMMES INFORMATIQUES DE CONTRÔLE DE L'INSTRUMENTATION DE SUSCEPTIBILITÉ MAGNÉTIQUE

**R. Provencher**
**Laboratoire de chimie**

**P. Bédard**
**Laboratoire de métallurgie extractive**

**avril 1989**

**Projet:** 30.52.05
**Impuretés dans le raffinage du cuivre**

MICROMEDIA

35 pp

**LABORATOIRES DES SCIENCES MINÉRALES**
**RAPPORT DE DIVISION LSM 89-44 (TR)**

Robert Provencher
Pierre Bédard

# PROGRAMMES INFORMATIQUES DE CONTRÔLE DE L'INSTRUMENTATION DE SUSCEPTIBILITÉ MAGNÉTIQUE

Énergie, mines et ressources
Laboratoire MSL
Ottawa
14 avril 1989

## PRÉSENTATION DES PROGRAMMES

Le contrôle du montage en susceptibilité magnétique se fait à partir de deux programmes principaux: SUSGW2.BAS compilé sous le nom HYST.EXE qui permet, entre autre, de traçer les courbes hystérèses et HPACQUG2.BAS compilé en SUSTEMP.EXE. Ce dernier permet d'acquérir des données en temps et en température et comporte également une boucle qui met les données en mémoire à chaque fois que 25 de celles-ci sont acquises. Il est à noter que les fichiers de ces programmes sont de type ASCII. Pour faciliter la compréhension du lecteur, les programmes ont été imprimés et placés à la suite de ce texte.

## PROGRAMME 1: SUSGW2.BAS

Le programme de susceptibilité se présente sous forme de menu où l'utilisateur peut choisir parmi les options disponibles. Si l'utilisateur fait une erreur lors de l'entrée d'une commande dans un menu, il peut revenir modifier cette commande en tapant U puis "return".

Le menu principal:

1) INTERACTIF
2) SUSCEPTIBILITÉ
3) SEND GRAPH TO PLOTTER
4) SAVE DATA ON HARD DISK
5) READ DATA ON HARD DISK
6) SEND DATA TO PRINTER OR ON SCREEN
7) CALL CURVE TO FITTING
8) CALL CORRECTION
9) ANNUNCIATOR
10) HP ACQUISITION
11) FOR ENDING
12) CALCUL OF SUSCEP, RETEN AND COERC

Les sous-menus:

1) INTERACTIF

Voir LECTURE OU ENTRÉE D'UNE COMMANDE dans le programme HPACQUG2.BAS.

2) SUSCEPTIBILITÉ

Le sous-menu ci-haut mentionné permet de traçer les courbes hystérèses. L'utilisateur doit y entrer la masse de l'échantillion, le moment et le champ maximaux du graphique à l'écran, le nombre de données désirées (500 au maximum), l'échelle du gaussmètre et du magnétomètre, le temps entre chaque mesure et finalement le nombre de segments. Ensuite, chaque segment demandé apparaîtra à l'écran et pourra être approuvé ou modifié. La dernière étape avant le début de l'acquisition des données est le choix d'une descente automatique du champ magnétique (Y pour oui et N pour non) et d'une valeur finale de ce champ.

3) SEND GRAPH TO PLOTTER

L'envoi de la courbe hystérèse à la traçeuse s'exécute assez facilement. Il s'agit d'entrer le nom du fichier correspondant à la courbe désirée, le nom de l'utilisateur, les échelles maximales du champ et du moment magnétique, le nombre de données, le saut de lecture, un commentaire et de choisir P si on veut que la courbe soit traçée en points et L si on préfère une ligne continue. Si vous avez beaucoup de données, il se pourrait que l'ordinateur vous envoie un message vous demandant d'attendre qu'il ait fini de traiter les données présentes dans son "buffer". Ensuite, vous devez appuyer sur "return".

4) SAVE DATA ON HARD DISK

Voir MISE EN MÉMOIRE DES DONNÉES dans le programme HPACQUG2.BAS.

5) READ DATA ON HARD DISK

Voir LECTURE DES DONNÉES dans le programme HPACQUG2.BAS.

6) SEND DATA TO PRINTER OR ON SCREEN

Voir LISTE DES DONNÉES (I/E) dans le programme HPACQUG2.BAS et remplacer I par P et E par S lors du choix entre l'imprimante ou l'écran.

7) CALL CURVE TO FITTING

8) CALL CORRECTION

Ces sous-menus (7 et 8) sont désactivés.

9) ANNUNCIATOR

Ce sous-menu permet d'ouvrir (normal ou inverse) et de fermer le circuit de l'électro-aimant. On peut aussi entrer et lire des commmandes interactives.

1 ON +

3 ON -

6 OFF

10) HP ACQUISITION

Voir le programme HPACQUG2.BAS.

11) FOR ENDING

Fin du programme susceptibilité.

12) CALCUL OF SUSCEP, RETEN AND COERC

Une courbe de susceptibilité de 500 données donne accès à ce sous-menu. Ce dernier offre la possibilité d'obtenir sur l'écran ou à l'imprimante: le nom du fichier, la masse de l'échantillon, le moment maximal, minimal et moyen, la retentivité et la coercivité

PROGRAMME 2: HPACQUG2.BAS

Le programme d'acquisition des données est construit sous forme de menus. Voici un résumé des options et possibilités offertes par chacun des sous-menus du menu principal.

Le menu principal:

1) LECTURE OU ENTRÉE D'UNE COMMANDE
2) TEMPÉRATURE DE RÉFÉRENCE
3) LECTURE DE T, EMU, OER
4) LISTE DES DONNÉES (I/E) (PRINT)
5) MISE EN MÉMOIRE DES DONNÉES (SAVE)
6) LECTURE DES DONNÉES (READ)
7) GRAPHIQUE À L'ÉCRAN
8) GRAPHIQUE À LA TRAÇEUSE
9) FIN

Les sous-menus:

1) LECTURE OU ENTRÉE D'UNE COMMANDE

A) Votre commande:

À ce stade l'utilisateur peut lire ou entrer sa commande. Celle-ci doit être écrite en lettres majuscules et suivie d'un nombre seulement si il s'agit d'une entrée.

Ex: MF 3000

Pour sortir de ce sous-menu, il faut taper 0 à la place de la commande interactive. L'ordinateur demande alors si vous avez terminé d'entrer ou de lire les commandes: Y si oui et N pour non.

B) Magnétomètre = 9 - HP = 5 quelle adresse ?

L'adresse correspond au numéro d'interface:

| | | |
|---|---|---|
| a) champ magnétique | 9 | MF |
| b) lecture du champ magnétique | 9 | OER |
| c) température | 5 | DCV4 |
| d) tension de référence | 5 | DCV3 |
| e) échelle du moment magnétique | 9 | MR |
| f) lecture du moment magnétique | 9 | EMU |

C) Votre commande demande-t-elle une réponse (Y/N) ?

Il faut inscrire Y pour une lecture de commande et N pour une entrée.

2) TEMPÉRATURE DE RÉFÉRENCE

A) Voltage de référence (volt) :

Quand on augmente ou diminue la température, il est important d'indiquer le voltage de départ; soit 0 au moment de l'élévation et le voltage final de la montée avant la diminution.

B) Champ magnétique (gauss)

3) LECTURE DE T, EMU, OER

A) Champ magnétique

B) Augmentation de temps:

C'est ici que l'on entre le temps en secondes qui s'écoulera entre chaque prise de données.

C) Augmentation de voltage:

On inscrit la différence de voltage entre chaque mesure (précédée d'un + ou d'un - selon qu'il s'agit d'une montée ou d'une descente). Pour rester à température constante pendant un certain temps, on doit écrire 0 comme augmentation de voltage.

D) Voltage initial (valeur lue du v.r.):

L'ordinateur inscrit ici la valeur de voltage qu'il a lu de la tension de référence en DCV3. L'utilisateur peut modifier cette valeur à son gré ou la conserver pour effectuer ses mesures.

E) Numéro de la mesure voulue (dernière mesure lue):

L'ordinateur indique par défaut la dernière mesure qu'il a lue. Si l'on désire continuer à acquérir des données à la suite des précédentes, on ne change pas le chiffre indiqué. Par contre, si l'on désire commencer une nouvelle série de mesures, il faut indiquer 0 à cet endroit.

F) Nombre de mesures avant le prochain arrêt:

Il est à noter que le programme n'accepte pas plus de 500 données au total.

Lorsque tous les ajustements ont été faits, l'acquisition des données commence et pour chacune de celles-ci l'ordinateur imprime à l'écran la température, le champ magnétique, le moment et la tension de référence. Une clé d'échappement permet d'interrompre l'acquisition des données en tout temps.

4) LISTE DES DONNÉES (I/E) (PRINT)

On choisit I pour l'imprimante et E pour l'écran. Dans les deux cas, on pourra obtenir le champ, la magnétisation, le voltage de référence et la température correspondants à chaque données.

5) MISE EN MÉMOIRE DES DONNÉES (SAVE)

Dans ce sous-menu, on demande à l'utilisateur si il veut tout d'abord voir la liste des fichiers déjà existants: Y pour oui et N pour non. Ensuite, apparaît sur l'écran le nombre de données enregistrées par l'ordinateur. Il s'agit dès lors d'indiquer combien de données on veut inscrire en mémoire et le saut de lecture entre chacune d'elles. Par exemple, si on veut une lecture globale, le saut sera 1.

6) LECTURE DES DONNÉES (READ)

Après avoir déterminé le fichier à lire, on inscrit le nombre de données de ce fichier et le saut de lecture ce qui permet de travailler sur ce fichier.

7) GRAPHIQUE À L'ÉCRAN

En ce qui concerne les paramètres pour l'acquisition des données, ce sont les mêmes que pour le numéro 2) avec en plus:

G) Échelle du magnétomètre (emu)

H) Axe des x: temps (1) ou température (2)

L'utilisateur a deux choix: faire des mesures en température ou en temps. Les mesures en temps servent à étudier le comportement dynamique des échantillons.

Les paramètres pour le graphique à l'écran sont les suivants:

A) Masse de l'échantillon

B) Échelle initiale (xi)

C) Échelle finale (xf)

D) Échelle initiale (yi)

E) Échelle finale (yf)

F) Graphique 1/susceptibilité (Y/N):

Il faut faire attention à la division par zéro.

L'ajustement de tous les paramètres étant fait, on peut maintenant observer l'acquisition des données sous forme de points sur l'écran et lire en haut de ce dernier la magnétisation, le voltage de référence et la température pour chacune des données qui apparaissent à l'écran. On peut cesser l'acquisition des données en tout temps pour modifier un paramètre en utilisant la clé d'échappement.

8) GRAPHIQUE À LA TRAÇEUSE

Dans ce sous-menu. il s'agit d'indiquer le nom du fichier, les échelles initiales et finales, le nombre de données et le saut de lecture pour que soit traçée la courbe du fichier.

9) FIN

Ce choix ramène l'utilisateur au menu de susceptibilité.

## CONCLUSION

Ces programmes assurent à l'utilisateur un contrôle presque parfait de tout le montage: instrumentation, acquisition des données, commandes par interface, etc... Même un utilisateur peu familier avec ce genre de système peut réussir à obtenir des résultats relativement satisfaisants.

Les auteurs remercient les étudiants du système coopératif qui ont collaborer à l'élaboration de ces programmes:

Sylvie Trépanier
Gwendoline Simon
Michel Héon
Claude Jean

LISTES DES
PROGRAMMES INFORMATIQUES

1) SUSGW2.BAS

2) HPACQUG2.BAS

3) FITTE.BAS

4) UTIL.BAS -sous-routines:

INTERACTIF
ATTEND
MENUE
OPENFILE
READDATA
SAVEDATA

```
        REM************************************************************************
        REM*                                                                      *
        REM*   Ce programme a ete redige par ROBERT PROVENCHER et PIERRE BEDARD   *
        REM*                                                                      *
        REM************************************************************************
        sw = 0: ym = .1: mx = 8000: nb = 500: ex = 10: time = 2.5: se = 5:stp=1
        DIM typ(20), rep$(20), que$(20)
        PRINT "allo"
        syscon% = 1
        I488addr% = &H318
        devaddr% = 1
        CALL INIT(syscon%, I488addr%, devaddr%)
        REM ******************** IBM SIMPLE STATIC INTERFACE ********************
        DIM px(510), py(510), x2(10), x1(10)
        cr$ = CHR$(13)
        CLS
        COLOR 1, 3, 1: CLS
        FOR I = 1 TO 7: PRINT " ": NEXT I
        PRINT TAB(15); " _____-
        PRINT "                *                                              *"
        PRINT " "; TAB(16); "* DEVICE MUST BE SET FOR CARRIAGE RETURN TERMINATOR *"
        PRINT TAB(15); " *_____*"

'       REM $INCLUDE: 'HPACQUG2.BAS'

        FOR I = 1 TO 5: PRINT " ": NEXT i: INPUT "              DATE : "; da$: CLS
        PRINT : PRINT : PRINT : PRINT : PRINT
        FOR e = 1 TO 6: PRINT : NEXT e: PRINT "          METTEZ CAPS LOCK ON"
        time% = -30000: CALL SETTIMER(time%)
'       a$ = "5": eoi$ = "10": per$ = "N": c$ = "RS": GOSUB 30710: c$ = "CLS12;OPN12": GOSUB 30710
'       vaf = 0: value0% = 0: GOSUB 30991
30380   REM ******************************************************
        REM ********************* MAIN LOOP **********************
        REM ********************** MENU ************************
        CLS : BEEP: PRINT : PRINT "         MAKE YOUR CHOICE...PRESS UP KEY TO STOP ANYWHERE": PRINT : PRINT
        KEY(11) ON: ON KEY(11) GOSUB 40000
        PRINT "          1- INTERACTIF"
        PRINT "          2- SUSCEPTIBILITE"
        PRINT "          3- SEND GRAPH TO PLOTTER"
        PRINT "          4- SAVE DATA ON HARD DISK "
        PRINT "          5- READ DATA ON HARD DISK "
        PRINT "          6- SEND DATA TO PRINTER OR ON SCREEN"
        PRINT "          7- CALL CURVE TO FITTING"
        PRINT "          8- CALL CORRECTION"
        PRINT "          9- ANNUNCIATOR"
        PRINT "          10- HP ACQUISATION"
        PRINT "          11- FOR ENDING"
        PRINT "          12- CALCUL OF SUSCEP,RETEN AND COERC"
        PRINT : PRINT : PRINT
```

```
50          INPUT "          WHAT NUMBER DO YOU CHOOSE "; choix
51          SELECT CASE choix
52           CASE IS = 1
53                  CALL interactif
54           CASE IS = 2
55                  GOSUB 31050
56                  BEEP
57                  BEEP
58                  BEEP
59           CASE IS = 3
60                  GOSUB 32180
61           CASE IS = 4
62                  CALL savedata(px(), py(), sw, se, n$)
63           CASE IS = 5
64                  CALL readdata(px(), py(), sw, se, nb, st, n$)
65           CASE IS = 6
66                  GOSUB 32020
67           CASE IS = 7
68                  GOSUB 33050
69           CASE IS = 8
70                  GOSUB 33180
71                  CALL CORRECT
72           CASE IS = 9
73                  GOSUB 33210
74           CASE IS = 10
75                  GOSUB 33430
76           CASE IS = 11
77                  GOSUB 31010
78           CASE IS = 12
79                  CALL fitte(n$, px(), py(), sw, se, st, nb)
80           CASE ELSE
81          END SELECT
82  '       IF choix = 7 THEN CALL CURVE3
83  '       IF choix = 10 THEN CALL  HPACQUI2
84          GOTO 30380
85  30590   REM ********************* INTERACTIF *********************
86  30710   REM ********************* COMMUNICATION *********************
87          REM ********************* OUTPUT COMMAND *********************
88          listen$ = a$
89          CALL WRITES(listen$, eoi$, c$)
90          IF per$ = "N" THEN 30990
91          REM ********************* GET RESPONSES *********************
92          rdata$ = SPACE$(50)
93          talk$ = a$
94          listen$ = ""
95          CALL READS(talk$, listen$, eoi$, rdata$, status%)
96          aa = VAL(rdata$)
97  30990   RETURN
98  30991   REM ********************* COMMANDE SUR LA CARTE *********************
```

```
 99          REM ********************* OUTPUT VOLTAGE (LABSTAR) *********************
100          address% = &H310
101          value1% = ABS(vmf)
102          CALL dacsetup(address%)
103          CALL altout12s(value1%, value0%)
104  30993   RETURN
105  31010   REM ********************* FOR ENDING *********************
106          CLS : FOR i = 1 TO 10: PRINT : NEXT i: PRINT "                                    GOOD BYE !"
107          IF na$ <> "" THEN PRINT : PRINT TAB(38); na$
108          FOR i = 1 TO 9: PRINT : NEXT i: STOP: END
109  31050   REM ********************* SUSCEPTIBILITE *********************
110          CLS
111          FOR i = 1 TO 8
112          typ(i) = 1
113          NEXT i
114          note$ = "NOTE:  ENTER THE SAMPLE WEIGHT = 1 FOR  AN Y-SCALE IN EMU"
115          que$(1) = "SAMPLE WEIGHT (g)= "
116          rep$(1) = STR$(sw)
117          que$(2) = "MAX. SCALE OF MOMENT (EMU) (SCREEN GRAPH) = "
118          rep$(2) = STR$(ym)
119          que$(3) = "MAX. SCALE OF FIELD (GAUSS) (SCREEN GRAPH) = "
120          rep$(3) = STR$(mx)
121          que$(4) = "NUMBER OF DATA = "
122          rep$(4) = STR$(nb)
123          que$(5) = "GAUSSMETER SCALE (0.1,1.0 OR 10 KG ) = "
124          rep$(5) = STR$(ex)
125          que$(6) = "MAGNETOMETER SCALE (EMU ) = "
126          rep$(6) = STR$(ey)
127          que$(7) = "TIME = "
128          rep$(7) = STR$(time)
129          que$(8) = "NUMBER OF SEGMENT = "
130          rep$(8) = STR$(se)
131          NBRQUES = 8
132          CALL menue(note$, que$(), rep$(), typ(), NBRQUES)
133          sw = VAL(rep$(1)): ym = VAL(rep$(2)): mx = VAL(rep$(3)): nb = VAL(rep$(4))
134          ex = VAL(rep$(5)): ey = VAL(rep$(6)): time = VAL(rep$(7)): se = VAL(rep$(8))
135          a9$ = "9": eoi$ = "13": per$ = "N": c$ = "MR " + STR$(ey): GOSUB 30710
136          IF sw = 0 THEN sw = 1
137          tg = 0
138          typ(1) = 1
139          OPEN "c:\vsm\data\parmsus" FOR INPUT AS #1
140          INPUT #1, st, sy
141          FOR i = 1 TO sy
142          INPUT #1, x1(i), x2(i)
143          typ(2 * i) = 1
144          typ(2 * i + 1) = 1
145          NEXT i
146          CLOSE #1
147  31280   FOR m = 1 TO se
```

```
148           que$(2 * m - 1) = "SEGMENT " + STR$(m) + "   FIELD MIN = "
149           rep$(2 * m - 1) = STR$(x1(m))
150           que$(2 * m) = "              FIELD MAX = "
151           rep$(2 * m) = STR$(x2(m))
152           NEXT m
153           NBRQUES = 2 * se
154           CALL menue("", que$(), rep$(), typ(), NBRQUES)
155           FOR m = 1 TO se
156           x1(m) = VAL(rep$(2 * m - 1))
157           x2(m) = VAL(rep$(2 * m))
158           tg = tg + ABS(x2(m) - x1(m))
159           NEXT m
160           OPEN "c:\vsa\data\paramsus" FOR OUTPUT AS #1
161           PRINT #1, tg / (nb - 1), se
162           FOR m = 1 TO se
163           PRINT #1, x1(m), x2(m)
164           NEXT m
165           CLOSE #1
166           que$(1) = "AUTOMATIC LOWERING OF FIELD <Y/N> "
167           op$ = "Y"
168           rep$(1) = op$
169           typ(1) = 0
170           que$(2) = "FINAL FIELD VALUE "
171           rep$(2) = STR$(last)
172           typ(2) = 1
173           NBRQUES = 2
174           CALL menue("", que$(), rep$(), typ(), NBRQUES)
175           op$ = rep$(1): last = VAL(rep$(2))
176           REM ********** LECTURE DES DONNEES **********
177           CLS
178           SCREEN 2
179           LINE (320, 25)-(320, 175)
180           LINE (0, 100)-(640, 100)
181   31490   j = 0
182           k2 = x1(1)
183           FOR i = 1 TO se
184           GOSUB 33370
185           k = k2: GOSUB 33120
186           st = SGN(x2(i) - x1(i)) * (tg / (nb - 1)): xi = k2: xf = x2(i): GOSUB 31650
187           NEXT i
188           REM ********** REMISE DU CHAMP A ZERO **********
189   31560   IF op$ <> "Y" THEN 31645
190           vwf = 0: GOSUB 30991
191           temps = 7.5
192           CALL attend(temps)
193           a9$ = "5": soi$ = "10": per$ = "N": c$ = "CLS12;OPN12": GOSUB 30710
194           GOSUB 31970
195   31645   RETURN
196   31650   REM ********** MESURE DE EMU ET OER **********
```

```
197           FOR k = xi TO xf STEP st
198           j = j + 1
199           vmf = INT(ABS(k) * .4095)
200           GOSUB 30991
201           CALL attend(time): a9$ = "9": eol$ = "13"
202           per$ = "Y": c$ = "EMU ": GOSUB 30710
203           MOMENT = aa: py(j) = MOMENT
204           c$ = "OER ": GOSUB 30710
205           CHAMP = aa: px(j) = CHAMP: IF ex = .1 THEN px(j) = CHAMP / 3.1623:     CHAMP = px(j)
206           VIEW PRINT 1 TO 24
207           PRINT USING "NO = ####                    PRESS DOWN KEY TO STOP "; j
208           PRINT USING "CHAMP = #####  GAUSS              MOMENT = ##.##^^^^ EMU"; CHAMP; MOMENT
209           PRINT USING "MF    = #####  GAUSS"; INT(10 * k) / 10
210           zz$ = "Y"
211           x% = INT(CHAMP * 320 / mx + 320)
212           y% = INT(100 - (MOMENT * 75 / ym))
213           y% = ABS(y%)
214           DRAW "BM=" + VARPTR$(x%) + ",=" + VARPTR$(y%)
215           DRAW "M=" + VARPTR$(x%) + ",=" + VARPTR$(y%)
216           KEY(14) ON: ON KEY(14) GOSUB 31950
217           k2 = k + st
218 31930   NEXT k
219           RETURN
220 31950   k = xf: i = se
221           PRINT " I AM WORKING "
222           RETURN
223 31970   REM ********************* FIN DE MESURE DE EMU ET OER **********************
224 31980   PRINT "*** PRESS RETURN TO CONTINUE ***": BEEP: BEEP
225           INPUT A$
226           SCREEN 0, 0, 0: COLOR 1, 3, 1
227           IF O$ <> "ERR" THEN 32010
228           GOSUB 31010
229 32010   RETURN
230 32020   REM ********************* PRINT ROUTINE **********************
231           CLS
232           PRINT : INPUT "NUMBER OF DATA = "; nb: PRINT
233           PRINT : INPUT "NUMBER OF STEP = "; st: PRINT
234           CLS : PRINT : PRINT " CHOOSE <P> FOR PRINTER OR <S> FOR SCREEN ": INPUT A$:   IF A$ = "P" THEN 32090
235           PRINT : PRINT "     FILE NAME : "; n$: PRINT : PRINT "SAMPLE WEIGHT "; sw:    PRINT
236           PRINT "  CHAMP (GAUSS) MOMEMT (EMU) ": PRINT : GOTO 32110
237 32090   LPRINT : LPRINT "      FILE NAME : "; n$: LPRINT : LPRINT "SAMPLE WEIGHT "; sw: LPRINT
238           LPRINT "CHAMP (GAUSS) MOMENT (EMU) ": LPRINT
239 32110   FOR i = 1 TO nb STEP st
240           IF A$ = "S" THEN PRINT i; TAB(10); px(i); TAB(20); py(i); TAB(35); (px(i) - px(i - 1))
241           IF A$ = "P" THEN LPRINT i; TAB(10); px(i); TAB(20); py(i); TAB(35); (px(i) - px(i - 1))
242           NEXT i
243           PRINT
244           PRINT "*** PRESS RETURN TO CONTINUE ***": INPUT A$
245 32170   RETURN
```

```
246  32180   REM ********************* SEND GRAPH TO HIPLOT *********************
247          note$ = "                SEND GRAPHIC TO PLOTTER"
248          ques(1) = "FILE NAME :"
249          rep$(1) = n$
250          ques(2) = "OPERATOR NAME : "
251          rep$(2) = na$
252          ques(3) = "MAX SCALES X <GAUSS> = "
253          ques(4) = "MAX SCALES Y <EMU> = "
254          rep$(3) = STR$(mx)
255          rep$(4) = STR$(my)
256          ques(5) = "NUMBER OF DATA = "
257          rep$(5) = STR$(nb)
258          ques(6) = "NUMBER OF STEPS IN THE READING = "
259          rep$(6) = STR$(st)
260          ques(7) = "COMMENTS :"
261          IF comment$ = "" THEN comment$ = "NIL"
262          rep$(7) = comment$
263          ques(8) = "POINTS OR LINE <P/L> "
264          tg$ = "L"
265          rep$(8) = tg$
266          FOR i = 3 TO 6
267          typ(i) = 1
268          NEXT i
269          typ(1) = 0: typ(2) = 0: typ(7) = 0: typ(8) = 0
270          NBRQUES = 8
271          CALL menue(note$, ques$(), rep$(), typ(), NBRQUES)
272          n$ = rep$(1): na$ = rep$(2)
273          mx = VAL(rep$(3)): my = VAL(rep$(4)): nb = VAL(rep$(5))
274          st = VAL(rep$(6)): comment$ = rep$(7): tg$ = rep$(8)
275          IF sw = 0 THEN sw = 1
276          INPUT "READY <Y/N> "; tp$
277          IF tp$ = "N" THEN 32460
278          a9$ = "30": eoi$ = "10": per$ = "N"
279          c$ = ";:ECM": GOSUB 30710
280          c$ = ";:H": GOSUB 30710
281          c$ = "U A 0 800 D 2000 800": GOSUB 30710
282          c$ = "U A 1000 1600 D 1000 0 U": GOSUB 30710
283          GOSUB 32810
284          c$ = "A ": GOSUB 30710
285          PRINT : PRINT : INPUT "JE REPRENDS MON SOUFFLE ....HIT RETURN"; S$
286          FOR i = 1 TO nb STEP st
287          x% = INT(px(i) * 1000 / mx) + 1000
288          y% = INT(py(i) * 800 / my) + 800
289          IF tg$ = "P" THEN
290          c$ = "U " + STR$(x%) + " " + STR$(y%) + " D "
291          GOSUB 30710
292          ELSE
293          c$ = " " + STR$(x%) + " " + STR$(y%) + " D "
294          GOSUB 30710
```

```
295           END IF
296           NEXT I
297           c$ = ";:CS": GOSUB 30710
298           c$ = ";:H": GOSUB 30710
299           c$ = ";:@": GOSUB 30710
300  32460    RETURN
301  32810    REM ******************** AXES ROUTINE **********************
302           my = (INT((my / sw) * 100000!)) / 100000!
303           c$ = "U A 975 1600 D 1025 1600 ":                                          GOSUB 30710
304           c$ = "U A 1075 1575 S(S4,NI,GO,X10,YO) " + STR$(my) + "_":                 GOSUB 30710
305           c$ = "U A 975 1200 D 1025 1200 ":                                          GOSUB 30710
306           c$ = "U A 975 400 D 1025 400 ":                                            GOSUB 30710
307           c$ = "U A 975 0 D 1025 0 ":                                                GOSUB 30710
308           c$ = "U A 1100 0 S(S4,NI,GO,X10,YO) " + STR$(-my) + "_":                   GOSUB 30710
309           c$ = "U A 0 775 D 0 825 ":                                                 GOSUB 30710
310           c$ = "U A 0 725 S(S4,NI,GO,X100,YO)-" + STR$(mx) + "_":                    GOSUB 30710
311           c$ = "U A 500 775 D 500 825 ":                                             GOSUB 30710
312           c$ = "U A 1500 775 D 1500 825 ":                                           GOSUB 30710
313           c$ = "U A 2000 775 D 2000 825 ":                                           GOSUB 30710
314           c$ = "U A 1925 725 S(S4,NI,GO,X10,YO) " + STR$(mx) + "_":                  GOSUB 30710
315           c$ = "U A 1300 725 S(S4,NI,GO,X10,YO) MAG. FIELD ( GAUSS )_":              GOSUB 30710
316           c$ = "U A 950 1200 S(S4,NI,GO,XO,Y10) MOMENT ( EMU/g )_":                  GOSUB 30710
317           c$ = "U A 0 1750 S(S4,NI,GO,X10,YO) " + "DATA FILE NAME : " + n$ + " _":   GOSUB 30710
318           c$ = "U A 0 1700 S(S4,NI,GO,X10,YO) DATE : " + da$ + " _":                 GOSUB 30710
319           c$ = "U A 0 1650 S(S4,NI,GO,X10,YO) COMMENT : " + comment$ + " _":          GOSUB 30710
320           c$ = "U A 1000 1750 S(S4,NI,GO,X10,YO) OPERATOR : " + na$ + " _":          GOSUB 30710
321           c$ = "U A 1000 1700 S(S4,NI,GO,X10,YO) SAMPLE WEIGHT : " + STR$(sw) + " (g) _": GOSUB 30710
322           'c$ = "U A 500 1900 S(S4,NI,GO,X10,YO) Radiation and Mineral Physics " + " _":  GOSUB 30710
323           my = my * sw
324  33040    RETURN
325  33050    REM ********************* CALL CURVE 2 POUR LISSAGE *********************
326           CLS
327           PRINT : PRINT : PRINT "*** ATTENTION SAVE YOUR DATA BEFORE CONTINUE *** ": PRINT
328           PRINT : PRINT " MAIN MENU <Y/N> ": INPUT A$
329           IF A$ = "Y" THEN 33110
330  33110    RETURN
331  33120    REM ********************* ATTENTE APRES LE SEGMENT *********************
332           vmf = INT(ABS(k) * .4095): GOSUB 30991
333           temps = 1.5
334           CALL attend(temps)
335  33160    RETURN
336  33180    REM ********************* CORRECTION *********************
337           CLS : PRINT : PRINT : PRINT " **** CALL CORRECTION ****"
338  33182    RETURN
339  33210    REM ********************* ANNUNCIATOR *********************
340           CLS
341  33255    per$ = "N"
342           PRINT : PRINT : PRINT : PRINT : PRINT "                I = 6  OFF  ": PRINT :   PRINT "                I =
     1   MAIN +": PRINT :    PRINT "                I = 3   MAIN -": PRINT : INPUT I
```

```
343           GOSUB 33370: CALL interactif
344           PRINT : PRINT : INPUT "                    *** END OF SETTING  <Y/N> ***"; A$
345           IF A$ = "Y" THEN 33315
346           GOTO 33255
347  33315   RETURN
348  33330   REM ********************* ERROR SUBROUTINE *********************
349           a9$ = "5": eoi$ = "10": per$ = "N": c$ = "CLS12;OPN12": GOSUB 30710
350           PRINT : PRINT : PRINT
351           PRINT :  PRINT "                  ****** ERROR ******  ":      O$ = "ERR": GOSUB 31980
352           RETURN
353  33370   REM ********************* ANNUNCIATOR *********************
354           per$ = "N"
355           IF i = 1 THEN
356           a9$ = "5"
357           eoi$ = "10"
358           c$ = "CLS10;OPN10 "
359           GOSUB 30710
360           END IF
361           IF i = 3 THEN
362           a9$ = "5"
363           eoi$ = "10"
364           GOSUB 45000
365           c$ = "CLS11;OPN11"
366           GOSUB 30710
367           END IF
368           IF i = 5 THEN
369           a9$ = "5"
370           eoi$ = "10"
371           GOSUB 45000
372           c$ = "CLS10;OPN10"
373           GOSUB 30710
374           END IF
375           IF i = 6 THEN
376           a9$ = "5"
377           eoi$ = "10"
378           GOSUB 45000
379           END IF
380           RETURN
381  33430   REM ********************* CALL HPACQUISITION *********************
382           CLS : PRINT : PRINT : PRINT "CALL HP-ACQUISITION"
383  33450   RETURN
384  33460   REM ********************* CALCUL DE SUSCEPTIBILITE *********************
385  '        note$ = "Calcul de suseptibilite"
386  '        que$(1) = "NUMBER OF SEGMENTS = "
387  '        rep$(1) = STR$(se)
388  '        que$(2) = "NUMBER OF DATA = "
389  '        rep$(2) = STR$(nb)
390  '        que$(3) = "DATA FILE NAME : "
391  '        rep$(3) = n$
```

```
392  '        que$(4) = "SAMPLE WEIGHT = "
393  '        rep$(4) = STR$(sw)
394  '        FOR i = 1 TO 4
395  '        typ(i) = 1      ' 1 indique a (menue) que les valeurs a obtenir
396  '        NEXT i            ' sont numerique
397  '        typ(3) = 0
398  '        CALL menue(note$, que$(), rep$(), typ(), 4)
399  '        se = VAL(rep$(1)): nb = VAL(rep$(2)): n$ = rep$(3): sw = VAL(rep$(4))
400  '        OPEN "C:\VSW\DATA\LIEN" FOR OUTPUT AS #1
401  '        PRINT #1, nb: PRINT #1, se: PRINT #1, sw
402  '        PRINT #1, n$ + cr$
403  '        CLOSE #1
404  33600    RETURN
405  40000    REM ********** POUR ARRETER PARTOUT **********
406  '        IF choix = 1 THEN RETURN 30705
407           IF choix = 2 THEN RETURN 31645
408           IF choix = 3 THEN RETURN 32460
409  '        IF choix = 4 THEN RETURN 32610
410  '        IF choix = 5 THEN RETURN 32790
411           IF choix = 6 THEN RETURN 32170
412           IF choix = 7 THEN RETURN 33110
413           IF choix = 8 THEN RETURN 33182
414           IF choix = 9 THEN RETURN 33315
415  '        IF choix = 10 THEN RETURN 33450
416           IF choix = 12 THEN RETURN 33600
417           RETURN
418  45000    REM ********** OFF **********
419           c$ = "CLS12;OPN12"
420           GOSUB 30710
421           sec = 1
422           CALL attend(sec)
423           RETURN
424  45001    REM ********** STOP **********
425           phy = 1
426           RETURN
```

```
 1        REM************************************************************************
 2        REM*                                                                      *
 3        REM*  Ce programme a ete redige par ROBERT PROVENCHER ET PIERRE BEDARD    *
 4        REM*                                                                      *
 5        REM************************************************************************
 6   96   SUB HPACQUI2 STATIC
 7        DEFINT I-K
 8   97   REDIM vr(501), cd(501)
 9        REM ********** px=oer  py=emu ***********
10        SHARED px() , py() , phy , value1% , value0% , ey , a9$ , eoi$ , per$ , c$
11        DIM que$(20), rep$(20), typ(20)
12   98   CLS : PRINT : PRINT : PRINT "                                    HELLO !"
13   99   FOR i = 1 TO 10: PRINT : NEXT i
14        INPUT "         ENTRER LA DATE S.V.P "; da$
15  120   REM ********************* MENU *********************
16  122   CLS
17  125   PRINT : PRINT "                                  ***** MENU *****"
18  130   PRINT : PRINT "          LECTURE OU ENTREE D'UNE COMMANDE       CH=1"
19  140   PRINT : PRINT "          TEMPERATURE DE REFERENCE               CH=2"
20  150   PRINT : PRINT "          LECTURE DE T,EMU,OER                   CH=3"
21  160   PRINT : PRINT "          LISTE DES DONNEES (I/E)      (PRINT)   CH=4"
22  170   PRINT : PRINT "          MISE EN MEMOIRE DES DONNEES  (SAVE)    CH=5"
23  180   PRINT : PRINT "          LECTURE DES DONNEES          (READ)    CH=6"
24  190   PRINT : PRINT "          GRAPHIQUE A L'ECRAN                    CH=7"
25  200   PRINT : PRINT "          GRAPHIQUE A LA TRACEUSE                CH=8"
26  210   PRINT : PRINT "          FIN                                    CH=9"
27  220   PRINT : PRINT : INPUT "                          FAITES VOTRE CHOIX "; ch
28  230   ON ch GOSUB 250, 330, 440, 670, 830, 1000, 1180, 1420, 1680
29  245   GOTO 122
30  250   CLS
31  260   REM ********************* LECTURE OU ENTREE D'UNE COMMANDE *********************
32        note$ = "                    LECTURE OU ENTREE D'UNE COMMANDE"
33  265   que$(1) = "VOTRE COMMANDE ": c$ = "0": rep$(1) = c$: typ(1) = 0
34  270   que$(2) = "MAGNETOMETRE = 9 - HP = 5 QUELLE ADDRESSE ? "
35        a9$ = "9": rep$(2) = a9$: typ(2) = 0
36        que$(3) = "VOTRE COMMANDE DEMANDE-T-ELLE UNE REPONSE (Y/N) ? "
37        per$ = "Y": rep$(3) = per$: typ(3) = 0
38        NBRQUES = 3
39        CALL menue(note$, que$(), rep$(), typ(), NBRQUES)
40        c$ = rep$(1): a9$ = rep$(2): per$ = rep$(3)
41  266   IF c$ = "0" THEN 320
42  310   IF a9$ = "9" THEN eoi$ = "13" ELSE eoi$ = "10"
43  315   GOSUB 2000
44        PRINT "REPONSE = ";dd :INPUT A$
45  320   RETURN
46  330   REM ********************* TEMPERATURE DE REFERENCE *********************
47        PRINT : PRINT : PRINT
48        address% = &H310
49        INPUT "VOLTAGE DE REFERENCE (VOLT) = "; voltref
```

```
50        INPUT "CHAMP MAGNETIQUE (GAUSS) = ";MF
51        value1%=INT(MF*.4095)
52        value0% = INT(voltref * 409.5)
53        CALL dacsetup(address%)
54        CALL altout12s(value1%, value0%)
55  430   RETURN
56  435   REM ********************* TEMPERATURE DE REFERENCE *********************
57        PRINT : PRINT : PRINT
58        address% = &H310
59        value0% = INT(voltref * 409.5)
60        CALL dacsetup(address%)
61        CALL altout12s(value1%, value0%)
62        RETURN
63  440   REM ********************* LECTURE DE T,EMU,OER *********************
64        phy = 0 : ll = 0 : NBDEMES = 0
65        note$ = "                    PARAMETRES POUR L'ACQUISITION DE DONNEES"
66        que$(1) = "CHAMP MAGNETIQUE ": rep$(1) = STR$(MF): typ(1) = 1
67        que$(2) = "AUGMENTATION DE TEMPS ": rep$(2) = STR$(delayt): typ(2) = 1
68        que$(3) = "AUGMENTATION DE VOLTAGE ": rep$(3) = STR$(delayv): typ(3) = 1
69        a9$ = "5": eoi$ = "10": per$ = "Y"
70        c$ = "DCV3": GOSUB 2000: PRINT dd
71        que$(4) = "VOLTAGE INITIAL (VALEUR LUE DU V.R.)": rep$(4) = STR$(dd): typ(4) = 1
72        que$(5) = "NUMERO DE LA MESURE VOULUE (DERNIERE MESURE LUE) ": rep$(5) = STR$(l): typ(5) = 1
73        que$(6) = "NOMBRE DE MESURES AVANT LE PROCHAIN ARRET ": rep$(6) =STR$ (NBINIT): typ(6) = 1
74        NBRQUES = 6
75        CALL menue(note$, que$(), rep$(), typ(), NBRQUES)
76        MF = VAL(rep$(1)): delayt = VAL(rep$(2)): delayv = VAL(rep$(3))
77        voltref = VAL(rep$(4)): l = VAL(rep$(5)): NBINIT = VAL(rep$(6))
78        value1%=INT(MF*.4095)
79  450   CLS : PRINT : INPUT "APPUYER SUR RETURN POUR L'ACQUISITION DE DONNNEES (KEY 12 (GAUCHE) POUR ARRETER)"; a$
80  460   PRINT
81        KEY(12) ON: ON KEY(12) GOSUB 45001
82        IF phy=1 THEN 650
83  470   l = l + 1
84  480   a9$ = "5": eoi$ = "10": per$ = "Y"
85  550   c$ = "DCV3": GOSUB 2000: vr(l) = dd :REM ***** LECTURE DU VOLTAGE DE REFERENCE *****
86  560   c$ = "DCV4": GOSUB 2000: GOSUB 2300 : cd(l) = dd: REM ***** LECTURE DU THERMO-COUPLE *****
87  570   a9$ = "9": eoi$ = "13": per$ = "Y"
88  580   c$ = "OER": GOSUB 2000: px(l) = dd  :REM ***** LECTURE DU CHAMP MAGNETIQUE *****
89  590   c$ = "EMU": GOSUB 2000: py(l) = dd  :REM ***** LECTURE DU MOMENT MAGNETIQUE *****
90  600   PRINT : PRINT " TEMPERATURE              "; l; " = "; cd(l)
91  610   PRINT : PRINT " CHAMP MAGNETIQUE         "; l; " = "; px(l)
92  620   PRINT : PRINT " MOMENT                   "; l; " = "; py(l)
93  630   PRINT : PRINT " TENSION DE REFERENCE     "; l; " = "; vr(l)
94        PRINT
95        NBDEMES=NBDEMES+1: IF NBDEMES > NBINIT THEN 650
96        voltref=voltref+delayv:if voltref > 10.5 then 650
97        if voltref < 0 then voltref=0
98        GOSUB 435
```

```
 99        CALL attend(delayt)
100        GOTO 480
101  650   PRINT : PRINT " APPUYER SUR RETURN POUR CONTINUER ": INPUT a$
102  660   RETURN
103  670   REM ********** LISTE DES DONNEES SUR IMPRIMANTE OU A L'ECRAN **********
104  680   PRINT "IMPRIME SUR L'IMPRIMANTE OU A L'ECRAN <I/E> ": INPUT a$
105  690   IF a$ = "E" THEN 760
106  700   LPRINT : LPRINT n$; "  MASSE DE L'ECHANTILLON: "; sw; " gramme": LPRINT : LPRINT
107  710   LPRINT " DONNEE(#NO)     CHAMP (GAUSS)   MAGNETISATION (EMU)     V R        D C" "
108  725   FOR i = 1 TO I:LPRINT USING"     ####       ";i;:LPRINT USING"    ####   ";px(i);:LPRINT USING"          #.####
       ";py(i);:LPRINT USING"        #.###  ";vr(i);:LPRINT USING"####.##    ";cd(i):PRINT
109  740   NEXT i
110  750   GOTO 810
111  760   PRINT : PRINT n$; "  MASSE DE L'ECHANTILLON: "; sw; " gramme": PRINT : PRINT
112  770   PRINT " DONNEE(#NO)     CHAMP (GAUSS)   MAGNETISATION (EMU)     V R        D C"
113  785   FOR i = 1 TO I:PRINT USING"     ####       ";i;:PRINT USING"    ####   ";px(i);:PRINT USING"          #.####
    ";py(i);:PRINT USING"        #.#    ";vr(i);:PRINT USING"####.##    ";cd(i):PRINT
114  800   NEXT i
115  810   PRINT : PRINT " APPUYER SUR RETURN POUR CONTINUER ": INPUT a$
116  820   RETURN
117  830   REM ********** MISE EN MEMOIRE DES DONNEES **********
118  840   CLS
119  850   PRINT "MENU SAVE "
120  860   PRINT : PRINT "LISTE DES FICHIERS ? ": INPUT a$
121  870   PRINT : IF a$ = "Y" THEN FILES "c:\vsm\data\*.*"
122  880   CALL openfile(n$)
123  890   note$ = "            NOMBRE DE DONNEES LUES             : " + STR$(I)
124  900   que$(1) = "NOMBRE DE DONNEES                 : ": rep$(1) = STR$(nb): typ(1) = 1
125        que$(2) = "NOMBRE DE SAUTS DURANT LA LECTURE : ": rep$(2) = STR$(st): typ(2) = 1
126        NBRQUES = 2
127        CALL menue(note$, que$(), rep$(), typ(), NBRQUES)
128        nb = VAL(rep$(1)): st = VAL(rep$(2))
129  930   PRINT #1, nb, st
130  940   FOR i = 1 TO nb STEP st
131  950   PRINT #1, t, px(i), py(i) ,vr(i),cd(i)
132  960   NEXT i
133  970   PRINT #1, sw
134  980   CLOSE #1
135  990   RETURN
136  1000  REM ********** LECTURE DES DONNEES DANS LES FICHIERS **********
137        CALL readfil(n$)
138  1080  INPUT #1, nb, st
139  1090  I = nb
140  1100  FOR i = 1 TO nb STEP st
141  1110  INPUT #1, t, px(i), py(i) ,vr(i),cd(i)
142  1120  NEXT i
143  1130  INPUT #1, sw
144  1140  CLOSE #1
145  1150  PRINT : PRINT "NOMBRE DE DONNEES = "; nb: PRINT "NOMBRE DE SAUTS = "; st
```

```
146  1160  INPUT a$
147  1170  RETURN
148  1180  REM ********** GRAPHIQUE A L'ECRAN **********
149  1190  CLS
150        phy = 0 : ll = 0 : NBDEMES = 0
151        note$ = "                    PARAMETRES POUR L'ACQUISITION DE DONNEES"
152        que$(1) = "CHAMP MAGNETIQUE: ": rep$(1) = STR$(MF): typ(1) = 1
153        que$(2) = "AUGMENTATION DE TEMPS: ": rep$(2) = STR$(delayt): typ(2) = 1
154        que$(3) = "AUGMENTATION DE VOLTAGE: ": rep$(3) = STR$(delayv): typ(3) = 1
155        a9$ = "5": eoi$ = "10": per$ = "Y"
156        c$ = "DCV3": GOSUB 2000: PRINT dd
157        que$(4) = "VOLTAGE INITIAL (VALEUR LUE DU V.R.): ": rep$(4) = STR$(dd): typ(4) = 1
158        que$(5) = "NUMERO DE LA MESURE VOULUE (DERNIERE MESURE LUE): ": rep$(5) = STR$(I): typ(5) = 1
159        que$(6) = "NOMBRE DE MESURES AVANT LE PROCHAIN ARRET: ": rep$(6) = STR$(NBINIT): typ(6) = 1
160        que$(7) = "ECHELLE DU MAGNETOMETRE (EMU): ": rep$(7) = STR$(ey): typ(7) = 1
161        que$(8) = "AXE DES X: TEMPS (1) OU TEMPERATURE (2) ?": rep$(8) = STR$(axex): typ(8) = 1
162        NBRQUES = 8
163        CALL menue(note$, que$(), rep$(), typ(), NBRQUES)
164        MF = VAL(rep$(1)): delayt = VAL(rep$(2)): delayv = VAL(rep$(3))
165        voltref = VAL(rep$(4)): I = VAL(rep$(5)): NBINIT = VAL(rep$(6))
166        ey = VAL(rep$(7)): axex = VAL(rep$(8))
167        a9$ = "9":eoi$ = "13": per$ = "N": c$ = "MR " + STR$(ey): GOSUB 2000
168        value1%=INT(MF*.4095)
169        CLS
170  1200  note$ = "                         PARAMETRES DU GRAPHIQUE A L'ECRAN"
171        que$(1) = "MASSE DE L'ECHANTILLON          : ": rep$(1) = STR$(sw): typ(1) = 1
172        que$(2) = "ECHELLE INITIALE (XI)           : ": rep$(2) = STR$(xi): typ(2) = 1
173        que$(3) = "ECHELLE FINALE (XF)             : ": rep$(3) = STR$(xf): typ(3) = 1
174        que$(4) = "ECHELLE INITIALE (YI)           : ": rep$(4) = STR$(yi): typ(4) = 1
175        que$(5) = "ECHELLE FINALE (YF)             : ": rep$(5) = STR$(yf): typ(5) = 1
176        que$(6) = "GRAPHIQUE 1/SUSCEPTIBILITE (Y/N) : ": rep$(6) = zz$: typ(6) = 0
177        NBRQUES = 6
178        CALL menue(note$, que$() ,rep$() ,typ() ,NBRQUES)
179        sw = VAL(rep$(1)): xi = VAL(rep$(2)): xf = VAL(rep$(3))
180        yi = VAL(rep$(4)): yf = VAL(rep$(5)): zz$ = rep$(6)
181  1240  SCREEN 2
182  1250  y = 150 - ((-1 * yi) / (yf - yi)) * 150
183  1260  LINE (10, y)-(639, y)
184  1270  LINE (10, 0)-(10, 180)
185  1280  IF zz$ = "Y" THEN yy = yf: yf = 1 / yi: yi = 1 / yy
186  1290  KEY(13) ON: ON KEY(13) GOSUB 45001
187        IF phy = 1 THEN 1390
188        I = I + 1
189        a9$ = "5": eoi$ = "10": per$ = "Y"
190
191
192  c$ = "DCV3": GOSUB 2000: vr(I) = dd :REM ***** LECTURE DU VOLTAGE DE REFERENCE *****
193        c$ = "DCV4": GOSUB 2000: GOSUB 2300 : cd(I) = dd: REM ***** LECTURE DU THERMO-COUPLE *****
194        a9$ = "9": eoi$ = "13": per$ = "Y"
```

```
195         c$ = "OER": GOSUB 2000: px(I) = dd  :REM ***** LECTURE DU CHAMP MAGNETIQUE *****
196         c$ = "EMU": GOSUB 2000: py(I) = dd  :REM ***** LECTURE DU MOMENT MAGNETIQUE *****
197           IF (II MOD 25) <> 0 THEN 1295
198           N$="TRANSITI.TMP"
199           OPEN "C:\VSM\DATA\" + n$ FOR OUTPUT AS #1
200           PRINT #1, II, 1
201           FOR I = 1 TO II STEP 1
202           PRINT #1, t, px(I), py(I) ,vr(I),cd(I)
203           NEXT I
204           PRINT #1, sw
205           CLOSE #1
206
207 1295        WHILE(I>II)
208         II=II + 1
209 1300  y = py(II): IF zz$ = "Y" THEN y = 1 / y
210 1305  IF axax = 2 THEN x% = ((cd(II) - xi) / (xf - xi)) * 630 + 10
211 1310  IF axax = 1 THEN x% = ((II - xi) / (xf - xi)) * 630 + 10
212 1320  y% = 170 - ((y - yi) / (yf - yi)) * 170
213 1330  DRAW "BM=" + VARPTR$(x%) + ",=" + VARPTR$(y%)
214 1340  DRAW "M=" + VARPTR$(x%) + ",=" + VARPTR$(y%)
215 1360  VIEW PRINT 1 TO 24
216 1380  PRINT "      L= "; II ; "      T= "; cd(II); "      EMU= "; py(II);_
217         "      VR= "; vr(II);"          APPUYER SUR KEY 13 (DROITE) POUR ARRETER"
218         WEND
219         NBDEMES = NBDEMES+1: IF NBDEMES > NBINIT THEN 1390
220         voltref = voltref + delayv:if voltref > 10 then voltref = 10
221         if voltref < 0 then voltref = 0
222         GOSUB 435
223         CALL attend(delayt)
224         GOTO 1290
225 1390  INPUT a$:PRINT " APPUYER SUR RETURN POUR CONTINUER"
226 1400  SCREEN 2:SCREEN 0, 0, 0: COLOR 1, 3, 1
227 1410  RETURN
228 1420  REM ********************* GRAPHIQUE A LA TRACEUSE **********************
229         note$ = "                          GRAPHIQUE A LA TRACEUSE"
230         que$(1) = "NOM DU FICHIER          : ": rep$(1) = n$: typ(1) = 0
231         que$(2) = "ECHELLE INITIALE (XI)   : ": rep$(2) = STR$(xi): typ(2) = 1
232         que$(3) = "ECHELLE FINALE (XF)     : ": rep$(3) = STR$(xf): typ(3) = 1
233         que$(4) = "ECHELLE INITIALE (YI)   : ": rep$(4) = STR$(yi): typ(4) = 1
234         que$(5) = "ECHELLE FINALE (YF)     : ": rep$(5) = STR$(yf): typ(5) = 1
235         que$(6) = "NOMBRE DE DONNEES       : ": rep$(6) = STR$(nb): typ(6) = 1
236         que$(7) = "NOMBRE DE SAUTS         : ": rep$(7) = STR$(st): typ(7) = 1
237         NBRQUES = 7
238         CALL menue(note$, que$(), rep$(), typ(), NBRQUES)
239         n$ = rep$(1): xi = VAL(rep$(2)): xf = VAL(rep$(3)): yi = VAL(rep$(4))
240         yf = VAL(rep$(5)): nb = VAL(rep$(6)): st = VAL(rep$(7))
241 1470  ay = -(yi / (yf - yi)) * 1600
242 1480  IF yi >= 0 THEN ay = 100
243 1490  PRINT "AY="; ay
```

```
244  1500  by = ay - 25: cy = ay + 25: dy = ay - 75
245  1530  a9$ = "30": eoi$ = "10": per$ = "N"
246  1540  c$ = ";:ECM": GOSUB 2000
247  1550  c$ = ";:H": GOSUB 2000
248  1560  GOSUB 1820
249  1570  c$ = "A ": GOSUB 2000
250        PRINT : PRINT
251  1575  INPUT "UN MOMENT...MON BUFFER EST PLEIN...APPUYER SUR RETURN QUAND J'AURAI FINI"; OK$
252  1580  FOR i = 1 TO nb STEP st
253  1590  x% = INT((cd(i) - xi) * 1900 / (xf - xi) + .5) + 100
254  1600  IF yi >= 0 THEN y% = INT((py(i) - yi) * 1500 / (yf - yi) + .5) + 100
255  1610  IF yi < 0 THEN y% = INT((py(i) - yi) * 1600 / (yf - yi) + .5)
256  1620  c$ = "U " + STR$(x%) + " " + STR$(y%) + " D ": GOSUB 2000
257  1630  NEXT i
258  1640  c$ = ";:CS": GOSUB 2000
259  1650  c$ = ";:H": GOSUB 2000
260  1660  c$ = ";:@": GOSUB 2000
261  1670  RETURN
262  1820  REM ********** AXES ROUTINES **********
263  1830  c$ = "U A 0 " + STR$(by) + "D 0 " + STR$(cy): GOSUB 2000
264  1840  c$ = "U A 125 " + STR$(dy) + " S(S4,NI,GO,X10,YO) " + STR$(xi) + "_": GOSUB 2000
265  1850  c$ = "U A 0 " + STR$(ay) + "D 2000 " + STR$(ay): GOSUB 2000
266  1860  c$ = "U A 2000 " + STR$(by) + "D 2000 " + STR$(cy): GOSUB 2000
267  1870  c$ = "U A 1925 " + STR$(dy) + " S(S4,NI,GO,X10,YO) " + STR$(xf) + "_": GOSUB 2000
268  1880  c$ = "U A 1300 " + STR$(dy) + "S(S4,NI,GO,X10,YO) TEMPERATURE (K)_": GOSUB 2000
269  1890  c$ = "U A 50 1600 D 150 1600 ": GOSUB 2000
270  1900  c$ = "U A 170 1575 S(S4,NI,GO,X10,YO) " + STR$(yf) + "_": GOSUB 2000
271  1910  c$ = "U A 100 1600 D 100 0 U ": GOSUB 2000
272  1920  c$ = "U A 50 0 D 150 0 ": GOSUB 2000
273  1930  IF yi < 0 THEN c$ = "U A 170 0 S(S4,NI,GO,X10,YO) " + STR$(yi) + "_ ": GOSUB 2000
274  1940  IF yi >= 0 THEN c$ = "U A 35 125 S(S4,NI,GO,X10,YO) " + STR$(yi) + "_ ": GOSUB 2000
275  1950  c$ = "U A 35 1250 S(S4,NI,G 0,X0,Y10) MOMENT (EMU)_": GOSUB 2000
276  1960  c$ = "U A 25 1750 S(S4,NI,GO,X10,YO) " + "DATA FILE NAME: " + n$ + " _": GOSUB 2000
277  1970  c$ = "U A 25 1700 S(S4,NI,GO ,X10,YO) DATE : " + da$ + " _": GOSUB 2000
278  1980  c$ = "U A 1000 1750 S(S4,NI,GO,X10,YO) SAMPLE WEIGHT : " + STR$(sw) + " _": GOSUB 2000
279  1990  RETURN
280  2000  REM ********** COMMUNICATION **********
281  2070  listen$ = a9$
282  2080  CALL WRITES(listen$, eoi$, c$)
283  2110  rdata$ = SPACE$(50)
284  2120  talk$ = a9$
285  2130  listen$ = ""
286  2135  IF per$ = "N" THEN 2180
287  2140  CALL READS(talk$, listen$, eoi$, rdata$, status%)
288  2145  dd = VAL(rdata$)
289  2180  RETURN
290  2300  REM ********** CONVERSION MILLIVOLT-CELCIUS **********
291        A0 = 4.22421E-01 : A1 = 2.41306E+01 : A2 = 7.33610E-02
292        A3 = -3.75087E-04 : A4 = -4.56948E-04 : A5 = 2.01608E-05
```

```
293         A6 = -2.45945E-07
294         dd = dd*1000
295         dd = 26+ A0 + A1*dd + A2*dd^2 + A3*dd^3 + A4*dd^4 + A5*dd^5 +A6*dd^6
296         RETURN
297  1680   REM ********** FIN **********
298  1685   CLS
299  1690   FOR i = 1 TO 10: PRINT : NEXT i: PRINT "                                    BYE"
300  1695   END SUB
301
302
303
```

```
SUB centre (nb, x(1), y(1), posx(1), posy(1)) STATIC
DEFINT I
ymax = y(posx(1))
ymin = y(posx(2))
ymoyen = (ymin + ymax) / 2
FOR i = 1 TO nb
        y(i) = y(i) - ymoyen
NEXT i
CALL posxy(x(), y(), posx(), posy(), nb)
END SUB

DEFSNG I
SUB fitte (n$, x(1), y(1), sw, se, st, nb) STATIC
' CE PROGRAMME PERMET DE TROUVER LES VALEURS DE X POUR Y=0
' ET LES VALEURS DE Y POUR X=0
'
'
' CETTE PREMIERE SECTION CHARGE EN MEMOIRE LE FICHIER DEMANDE PAR LE
' MODULE <READFIL>.
'
SCREEN 2
CLS
DEFINT I-J
        DIM posx(5), posy(5), x0tot(5), y0tot(5)
        emmax = 0
        emmin = 0
        noenr = nb
        IF n$ = "" THEN
                CALL readfil(n$)
                noenr = 0
                INPUT #1, nb, st
                FOR i% = 1 TO nb STEP st
                        noenr = noenr + 1
                        INPUT #1, x(noenr), y(noenr)
                NEXT i%
                INPUT #1, sw, se
                CLOSE #1
        END IF
        FOR I = 1 TO nb STEP st
                IF y(I) > emmax THEN emmax = y(I): magmax = x(I)
                IF y(I) < emmin THEN emmin = y(I): MAGMIN = x(I)
        NEXT I
        CLS
        VIEW (1, 1)-(300, 75), , 1
        WINDOW (-1, -1)-(1, 1)
        LINE (0, 1)-(0, -1)
        LINE (1, 0)-(-1, 0)
        FOR I = 1 TO nb - 2 STEP 2
                LINE (x(I) / magmax, y(I) / emmax)-(x(I + 2) / magmax, y(I + 2) / emmax)
```

```
50        NEXT I
51        LOCATE 11, 1: PRINT "Actual graph in memory"
52        CALL posxy(x(), y(), posx(), posy(), noenr)
53        IF posx(1) = 0 OR posy(1) = 0 THEN
54              PRINT "***ERREUR*** Il n'y a pas de point qui traverse"
55              PRINT "             soit l'axe des X ou l'axe des Y !!!"
56              a$ = "Y"
57              INPUT "voulez vous centrer la courbe pour effectuer le calcul (Y/N) [Y]"; a$
58              IF a$ = "" THEN a$ = "Y"
59              IF a$ = "Y" OR a$ = "y" THEN
60                    CALL centre(nb, x(), y(), posx(), posy())
61                    emumax=0
62                    emumin=10000000
63                    FOR I = 1 TO nb
64                          IF y(I) > emumax THEN emumax = y(I): magmax = x(I)
65                          IF y(I) < emumin THEN emumin = y(I): MAGMIN = x(I)
66                    NEXT I
67                    VIEW (302, 1)-(637, 75), , 1
68                    WINDOW (-1, -1)-(1, 1)
69                    LINE (0, 1)-(0, -1)
70                    LINE (1, 0)-(-1, 0)
71                    FOR I = 1 TO nb - 2 STEP 2
72                          LINE (x(I) / magmax, y(I) / emumax)-(x(I + 2) / magmax, y(I + 2) / emumax)
73                    NEXT I
74                    LOCATE 11, 1: PRINT "Actual graph in memory                New graph in memory"
75              ELSE
76                    INPUT "HIT <<RETURN>> FOR CONTINUE", a$
77                    SCREEN 0
78                    COLOR 1, 3, 1: CLS
79                    EXIT SUB
80              END IF
81        ELSE
82              INPUT "Voulez vous centrer la courbe (Y/N) [N]"; a$
83              IF a$ = "" THEN a$ = "N"
84              IF a$ = "Y" OR a$ = "y" THEN
85                    CALL centre(nb, x(), y(), posx(), posy())
86                    FOR I = 1 TO nb
87                          IF y(I) > emumax THEN emumax = y(I): magmax = x(I)
88                          IF y(I) < emumin THEN emumin = y(I): MAGMIN = x(I)
89                    NEXT I
90                    VIEW (302, 1)-(637, 75), , 1
91                    WINDOW (-1, -1)-(1, 1)
92                    LINE (0, 1)-(0, -1)
93                    LINE (1, 0)-(-1, 0)
94                    FOR I = 1 TO nb STEP 2
95                          LINE (x(I) / magmax, y(I) / emumax)-(x(I + 2) / magmax, y(I + 2) / emumax)
96                    NEXT I
97                    LOCATE 11, 1: PRINT "Actual graph in memory                New graph in memory"
98              END IF
```

```
99          END IF
100         FOR i = 1 TO 5
101                 PRINT "                                                                    ";
102         NEXT i
103         LOCATE 12, 1
104         INPUT "VOUS VOULEZ <LISSER> SUR CONBIEN DE POINTS: "; nbrPoint
105   '
106   ' ICI ON EFFECTUE LES CALCULS POUR TROUVER LES POINTS D'INTERSECTIONS
107   ' SUR LES AXES
108   '
109         FOR j = 1 TO 2
110                 yOtot = 0
111                 xOtot = 0
112                 FOR i = 1 TO nbrPoint
113                 ' CACUL DE LA PENTE DE LA PREMIERE DROITE.
114                         m = (y(posx(j) + i) - y(posx(j) - i)) / (x(posx(j) + i) - x(posx(j) - i))
115                         IF m = 0 THEN m = .0001
116                 ' CALCUL DU POINT Y POUR X=0.
117                         x0 = -m * x(posx(j) - i) + y(posx(j) - i)
118                 ' CALCUL DE LA PENTE DE LA DEUXIEME DROITE.
119                         m = (y(posy(j) + i) - y(posy(j) - i)) / (x(posy(j) + i) - x(posy(j) - i))
120                         IF m = 0 THEN m = .0001
121                 ' CALCUL DU POINT X POUR Y=0.
122                         y0 = -y(posy(j) + i) / m + x(posy(j) + i)
123                         yOtot = yOtot + y0
124                         xOtot = xOtot + x0
125                 NEXT i
126                 yOtot(j) = yOtot / nbrPoint
127                 xOtot(j) = xOtot / nbrPoint
128         NEXT j
129         LPRINT TAB(30); "Radiation and Mineral Physics"
130         PRINT TAB(30); "Radiation and Mineral Physics"
131         LPRINT ""
132         LPRINT TAB(5); "File name : "; n$
133         PRINT TAB(5); "File name : "; n$
134         LPRINT TAB(5); "Sample weight : "; sw; " g"
135         PRINT TAB(5); "Sample weight : "; sw; " g"
136         LPRINT ""
137         LPRINT TAB(5); "Maximum moment  : "; emumax / sw; " EMU/g at   "; magmax; " Gauss"
138         PRINT TAB(5); "Maximum moment  : "; emumax / sw; " EMU/g at   "; magmax; " Gauss"
139         LPRINT TAB(5); "Minimum moment  : "; emumin / sw; " EMU/g at   "; MAGMIN; " Gauss"
140         PRINT TAB(5); "Minimum moment  : "; emumin / sw; " EMU/g at   "; MAGMIN; " Gauss"
141         LPRINT TAB(5); "Average moment  : "; (emumax - emumin) / (2 * sw); " EMU/g"
142         PRINT TAB(5); "Average moment  : "; (emumax - emumin) / (2 * sw); " EMU/g"
143         LPRINT ""
144         LPRINT TAB(5); "Coercitivity  + : "; yOtot(2); "Gauss"
145         PRINT TAB(5); "Coercitivity  + : "; yOtot(2); "Gauss"
146         LPRINT TAB(5); "Coercitivity  - : "; yOtot(1); "Gauss"
147         PRINT TAB(5); "Coercitivity  - : "; yOtot(1); "Gauss"
```

```
148         LPRINT ""
149         LPRINT TAB(5); "Retentivity + :"; xOtot(1) / sw; " EMU/g"
150         PRINT TAB(5); "Retentivity + :"; xOtot(1) / sw; " EMU/g"
151         LPRINT TAB(5); "Retentivity - :"; xOtot(2) / sw; " EMU/g"
152         PRINT TAB(5); "Retentivity - :"; xOtot(2) / sw; " EMU/g"
153         LPRINT ""
154         LPRINT TAB(5); "Retentivity : "; (ABS(xOtot(1) - xOtot(2))) / (2 * sw); " EMU/g"
155         PRINT TAB(5); "Retentivity : "; (ABS(xOtot(1) - xOtot(2))) / (2 * sw); " EMU/g"
156         LPRINT TAB(5); "Coercitivity : "; (ABS(yOtot(1) - yOtot(2))) / 2; " EMU/g"
157         PRINT TAB(5); "Coercitivity : "; (ABS(yOtot(1) - yOtot(2))) / 2; "EMU/g"
158         LPRINT CHR$(12)
159         LPRINT CHR$(12)
160         INPUT "HIT <<RETURN>> FOR CONTINUE", a$
161         SCREEN 0
162         COLOR 1, 3, 1: CLS
163 END SUB
164
165 DEFSNG I-J
166 SUB posxy (x(1), y(1), posx(1), posy(1), noenr) STATIC
167 DEFINT J-K
168 '
169 ' CETTE SECTION TROUVE LES DEUX POSITIONS DANS LE TABLEAU OU LE "X" ET LE "Y"
170 ' CHANGENT DE SIGNE.
171 ' LES VALEURS  POSX() ET POSY()  REPRESENTENT DES POINTEURS DANS LE TABLEAU
172 ' DES ENREGISTREMENTS. CES POINTEURS POINTENT LA POSITION OU LES VALEURS DE
173 ' X ET DE Y  CHANGENT DE SIGNE.
174 '
175         xopr = 0: xypr = 0
176         posx(1) = 0
177         posy(1) = 0
178         j = 1: k = 1
179         FOR x = 3 TO noenr   ' il est important de partir de 3 pour eviter de
180                                      ' localisar la premiere position au point 0.
181                 IF xopr + 10 < x THEN
182                         IF SGN(x(x)) <> SGN(x(x - 1)) THEN
183                                 posx(j) = x
184                                 j = j + 1
185                                 xopr = x
186                         END IF
187                 END IF
188                 IF xypr + 10 < x THEN
189                         IF SGN(y(x)) <> SGN(y(x - 1)) THEN
190                                 posy(k) = x
191                                 k = k + 1
192                                 xypr = x
193                         END IF
194                 END IF
195         NEXT x
196 END SUB
```

```
SUB attend (sec) STATIC
DEFINT A-L
DEFSNG S
s = TIMER + sec
WHILE TIMER < s
WEND
END SUB

DEFSNG A-L
SUB interactif STATIC
DIM que$(3), rep$(3), typ(3)
30590 REM ********** INTERACTIF ******************************************
      CLS
30610 note$ = "Commandes interactives"
      que$(1) = "ENTER COMMAND =  "
      rep$(1) = c$
      typ(1) = 0   ' type caractere
      que$(2) = "VOTRE COMMANDE NECESSITE UNE REPONSE <Y/N> "
      rep$(2) = per$
      typ(2) = 0
      que$(3) = "ENTREZ  5 POUR HP ET 9 POUR MAGNETOMETRE "
      rep$(3) = a9$
      typ(3) = 0
      NBRQUES = 3
      CALL menue(note$, que$(), rep$(), typ(), NBRQUES)
      c$ = rep$(1): per$ = rep$(2): a9$ = rep$(3)
      IF c$ = "0" THEN EXIT SUB: REM  RETURN MAIN LOOP
      CC$ = MID$(c$, 1, 2): IF CC$ = "MF" THEN C1$ = MID$(c$, 3, 9): c = VAL(C1$)
      IF CC$ = "MF" THEN
          vmf = INT(c * .4095)
          address% = &H310
          value1% = ABS(vmf)
          CALL dacsetup(address%)
          CALL altout12s(value1%, value0%)
          GOTO 30610
      END IF
      IF a9$ = "9" THEN eoi$ = "13" ELSE eoi$ = "10"
      listen$ = a9$
      CALL WRITES(listen$, eoi$, c$)
      IF per$ = "N" THEN 30610
      REM  GET RESPONSES
      rdata$ = SPACE$(50)
      talk$ = a9$
      listen$ = ""
      CALL READS(talk$, listen$, eoi$, rdata$, status%)
      aa = VAL(rdata$)
      IF per$ = "N" THEN 30610
      PRINT : PRINT : PRINT "RESPONSE = "; aa: INPUT a$
END SUB
```

```
50
51  SUB menue (notes$, question$(1), reponse$(1), tpe(1), NBRQUES) STATIC
52  CLS
53  PRINT notes$
54  posx = FIX(20 / NBRQUES)
55  FOR i = 1 TO NBRQUES
56      LOCATE posx * i + 2, 3: PRINT question$(i); "("; reponse$(i); ")
57  NEXT i
58  FOR i = 1 TO NBRQUES
59  10  LOCATE posx * i + 2, 5 + LEN(reponse$(i)) + LEN(question$(i)): INPUT " ", reponse$
60      IF (reponse$ = "u" OR reponse$ = "U") AND i > 1 THEN i = i - 2: GOTO 100
61      IF reponse$ = "" THEN GOTO 100
62      a = VAL(reponse$)
63      IF tpe(i) = 1 AND a = 0 THEN BEEP: reponse$ = STR$(a)
64      LOCATE posx * i + 2, 3: PRINT question$(i); "("; reponse$; ")
65      reponse$(i) = reponse$
66  100 NEXT i
67  END SUB
68
69  SUB openfile (n$) STATIC
70          CLS
71          PRINT "MENU READ ": PRINT
72          PRINT : INPUT "FILES DIRECTORY <Y/N> (Y)"; a$
73          PRINT
74          IF a$ <> "N" THEN FILES "c:\vsm\DATA\*.*"
75  32471   PRINT : INPUT "DATA FILE NAME = "; n$
76          OPEN "C:\VSM\DATA\" + n$ FOR APPEND AS #1
77          x = LOF(1)
78          CLOSE #1
79          a$ = "Y"
80          IF x > 0 THEN
81              BEEP
82              PRINT "LE FICHIER " + n$ + " EXISTE !!!"
83              INPUT "VOULEZ VOUS L'EFFACER (Y/N)"; b$
84              IF b$ = "Y" THEN a$ = "Y" ELSE a$ = "N"
85          END IF
86          IF a$ = "N" THEN GOTO 32471
87          OPEN "C:\VSM\DATA\" + n$ FOR OUTPUT AS #1
88  END SUB
89
90  SUB readdata (px(1), py(1), sw, se, nb, st, n$) STATIC
91  CALL readfil(n$)
92  INPUT #1, nb, st
93  FOR i = 1 TO nb STEP st
94      INPUT #1, px(i), py(i)
95  NEXT i
96  INPUT #1, sw, se
97  CLOSE #1
98  PRINT : PRINT "number of data = "; nb: PRINT "number of step in the file"
```

```
 99 PRINT : INPUT "hit <<return>> to continue", a$
100 END SUB
101
102 SUB readfil (n$) STATIC
103       CLS
104       PRINT "MENU READ ": PRINT
105       PRINT : INPUT "FILES DIRECTORY <Y/N> (Y)"; a$
106       PRINT
107       IF a$ <> "N" THEN FILES "c:\vsm\DATA\*.*"
108 90000 PRINT : INPUT "DATA FILE NAME = "; n$
109       OPEN "C:\VSM\DATA\" + n$ FOR APPEND AS #1
110       x = LOF(1)
111       CLOSE #1
112       a$ = "Y"
113       IF x = 0 THEN
114            BEEP
115            PRINT "LE FICHIER " + n$ + " N'EXISTE PAS!!!"
116            a$ = "N"
117            KILL "C:\VSM\DATA\"+n$
118       END IF
119       IF a$ = "N" THEN GOTO 90000
120       OPEN "C:\VSM\DATA\" + n$ FOR INPUT AS #1
121 END SUB
122
123 SUB savedata (px(1), py(1), sw, se, n$) STATIC
124 DIM rep$(2), que$(2), typ(2)
125 32470 REM ********** SAVE DATA ON DISK **********
126       CLS
127       CALL openfile(n$)
128       note$ = "SAVE DATA ON DISK"
129       que$(1) = "NUMBER OF DATA = "
130       nb = 500
131       rep$(1) = STR$(nb)
132       typ(1) = 1                    '" nb " valeur numerique
133       que$(2) = "NUMBER OF STEPS IN THE READING = "
134       st = 1
135       rep$(2) = STR$(st)
136       typ(2) = 1
137       NBRQUES = 2
138       CALL menue(note$, que$(), rep$(), typ(), NBRQUES)
139       nb = VAL(rep$(1)): st = VAL(rep$(2))
140       PRINT #1, nb, st
141       FOR i = 1 TO nb STEP st
142           PRINT #1, px(i), py(i)
143       NEXT i
144       PRINT #1, sw, se
145       CLOSE #1
146 END SUB
147
```